

## FP 15.7: A 0.25 $\mu$ m x86 Microprocessor with a 100MHz Socket 7 Interface.

R. Khanna, A. Ben-Meir, L. DiGregorio, D. Draper, R. Krishna,  
R. Maley, A. Mehta, S. Oberman, L. Tsai, T. Williams

Advanced Micro Devices, Inc., Milpitas, CA

The AMD-K6 3D MMX enabled processor is an enhanced follow-on to the AMD-K6 MMX enabled processor. [1] It is implemented in a 0.25 $\mu$ m 2.1V CMOS process to achieve a higher frequency of operation and 80 sq. mm die size (Table 1, Figure 6).

The K6-3D unit implements 19 floating-point (FP) vector operations to enhance 3D graphics and audio performance. The AMD-3D instructions operate in a SIMD fashion on 64b operands, similar to MMX instructions. Unlike MMX, the operands are two sets of 32b single-precision FP values rather than integers. The FP format is compatible with the IEEE 754 standard. However, only round-to-nearest-even is supported for most operations, and conversion results are truncated. All overflows are clamped to the maximum representable value. All numbers with a magnitude smaller than the smallest representable value are flushed to zero. NaNs and infinities are not supported, and instructions neither generate exceptions nor set status flags.

K6-3D instructions have a latency/throughput of 2/1 cycles, using three pipelined functional units (Figure 1). A non-multiply vector instruction and a vector multiply can be dispatched in each cycle, yielding a maximum of 4 FLOPS-per-cycle. The FP adder uses a split-significand datapath design based upon exponent range, with a compound integer adder in each datapath producing both the sum and sum+1. The use of compound adders allows final rounding and recomplementation of negative results to be performed by selection, rather than through an explicit incrementor. [2] The multiplier computes all FP multiplies required for AMD-3D and all integer multiplies required for MMX.

Three new instructions are added to the MMX execution unit. The unit executes two ALU operations per cycle or one ALU operation and either a shift or multiply operation for a sustained retirement rate of two instructions per cycle. Multiply operations are pipelined over two cycles. The unit does not impose instruction pairing restrictions on compilers or on the reorder buffer, instead resolving all dependencies itself at execution time. CPI on the Intel multimedia benchmark has been improved by over 20%.

Three key memory subsystem enhancements have been implemented to achieve higher performance, especially on systems that use the relatively slow 66MHz bus interface: pipelining of data reads and data writes on the pentium bus, speculative data cache fills, and the completion of store operations while a fill is in progress. These features combine to yield a simulated average cycles per instruction (CPI) improvement of 2 to 10%.

A key problem in achieving 100MHz bus operation is the greater fraction of the cycle time occupied by uncertainties in crossing between the processor clock (PCLK) and the bus clock (BCLK) domains, where PCLK frequency is an integer or half-integer multiple of BCLK. Domain crossing is inherent in the design of "loop-back paths" that consist of incoming signals whose assertion in a given cycle must trigger a corresponding external bus event in the very next BCLK cycle. The original K6 used a scheme in which the loop-back inputs are combined with internal signals, then latched by PCLK before being registered by BCLK for off-chip transmission (Figure 2a). The PCLK latch, in the case of an integer

PCLK to BCLK frequency ratio, is transparent during PCLK low to ensure that no hold-time problem occurs during coincident PCLK and BCLK rising edges. Placing the PCLK latch in the loop-back paths requires a clock domain crossing and the logic must be fast enough to compensate for clock uncertainties. This scheme achieves 100MHz bus operation by eliminating the PCLK latch and solving the hold-time problem by updating the PCLK register only on PCLK edges that are not coincident with a BCLK rising edge (Figure 2b, 2c).

The primary problem faced in the design of input/output (I/O) circuitry for the .25 $\mu$ m technology is that the gate oxide of the transistors cannot withstand more than 2.5V for long-term reliable operation, yet industry standards require a 3.3V interface for chip-to-chip signaling. A circuit configuration protects the gate oxides of the IO drivers and receivers. The alternative, to include transistors with thicker oxides in the fabrication process, would have increased the number of manufacturing steps and thus the cost of the product. Figure 3 shows the output driver circuitry with the protection scheme. To significantly speed up the level shifter, capacitors are added so that the pMOS gates couple up or down with the nMOS gate in the same stack. A bandgap voltage generator provides a constant voltage, independent of process, temperature, and VDDIO. This voltage is amplified, as shown in figure 4, to derive VREF2 and VREF1.

The PCLK clock buffer is implemented in three columns of drivers, each with 28 ribs whose outputs are all connected together to drive a mesh of M4 & M5 metal lines. Skew is further reduced by driving each rib with a local driver whose strength is programmed after base-layer tapeout by a contact mask to be proportional to its local load. Post-layout extraction and simulation of the clock network shows chip-wide clock skew to be less than 50ps.

The K6 design is based on a single-wire clocking scheme with edge-triggered flip-flops that have a positive hold-time requirement, and thus is susceptible to hold-time races. [3] To guarantee function, compliance to the inequality in Figure 5 is checked across all operating corners. The number of false violations is reduced by use of a clock skew equation to reduce worst-case skew between a violating pair of registers whose physical coordinates are known. Fixing violations is automated by an in-house CAD tool that examines all min-paths and max-path sensitivities along them to find where delays can be added without incurring a cycle time penalty.

The design is shown by extensive analysis to attain a lifetime of sustained performance of a minimum of ten years. Degradations from the aging mechanisms of electromigration and hot carrier injection (HCI) are considered. Even in the presence of 12% individual transistor degradation, percentage slowdown of partial paths due to HCI is less than 1.9%.

### References:

- [1] Draper, D., et al., "Circuit Techniques in a 266 MHz MMX-Enabled Processor," IEEE Journal of Solid-State Circuits, Nov., 1997.
- [2] Oberman, S., H. Al-Twaijry, M. J. Flynn, "The SNAP project: design of floating point arithmetic units," Proc. 13th IEEE Symp. Computer Arithmetic, pp. 156-165, July, 1997.
- [3] Partovi, H., et al., "Flow-Through Latch and Edge-Triggered Flip-Flop Hybrid Elements," ISSCC Digest of Technical Papers, pp. 138-139, Feb., 1996.



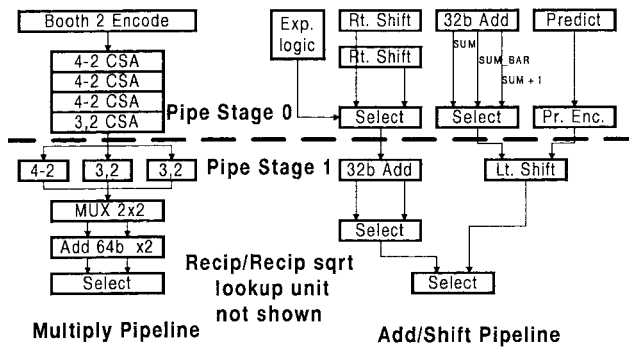


Figure 1: 3D Execution Units.

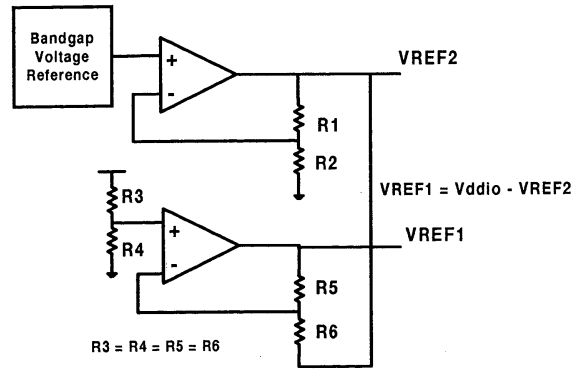


Figure 4: Reference voltage generation circuit.

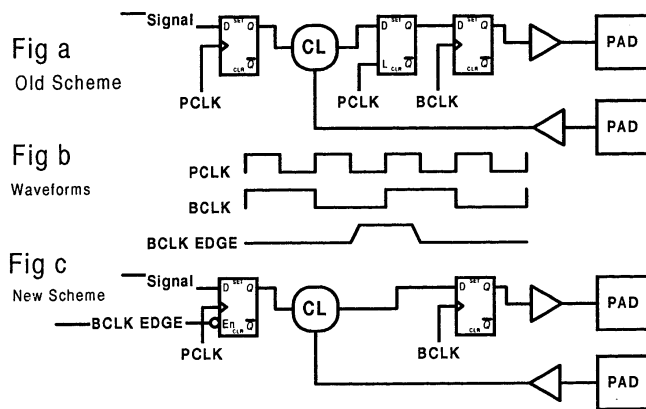


Figure 2: Loop-back path logic.

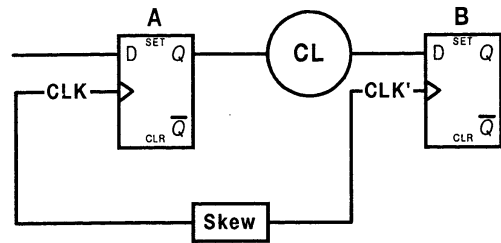


Figure 5: Clock vs. data rate.

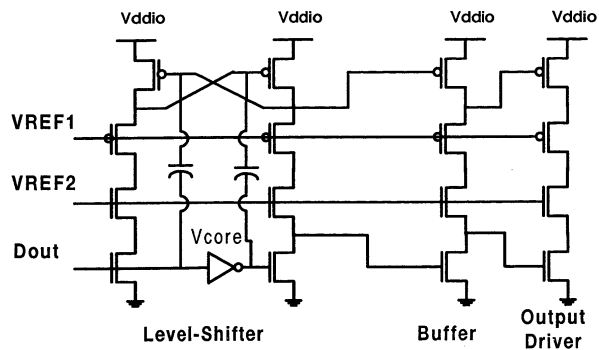


Figure 3: Protected output driver.

Gate oxide thickness	4.8nm
Polysilicon width/space	0.25/0.375nm
Local interconnect width/space	0.25/0.375nm
Metal 1 - 3 width/space	0.5/0.375nm
Metal 4 width/space	0.625/0.5nm
Metal 5 width/space	1.5/1.5nm

Table 1: 0.25µm process dimensions.

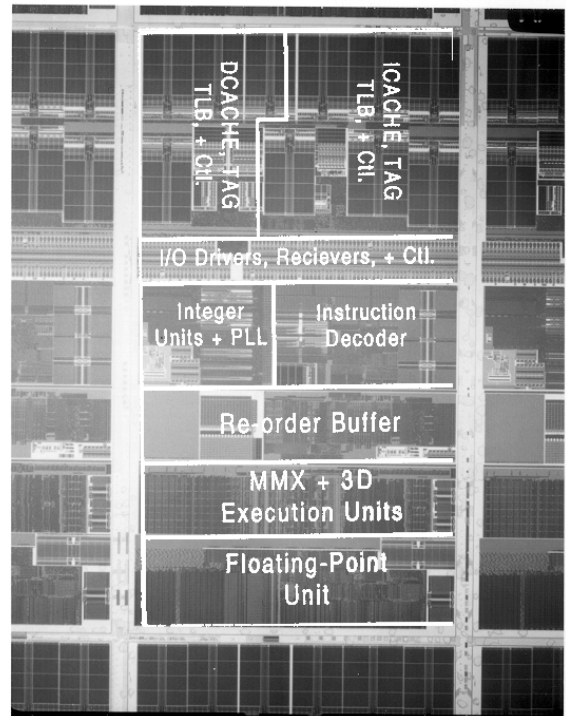


Figure 6: Die micrograph.

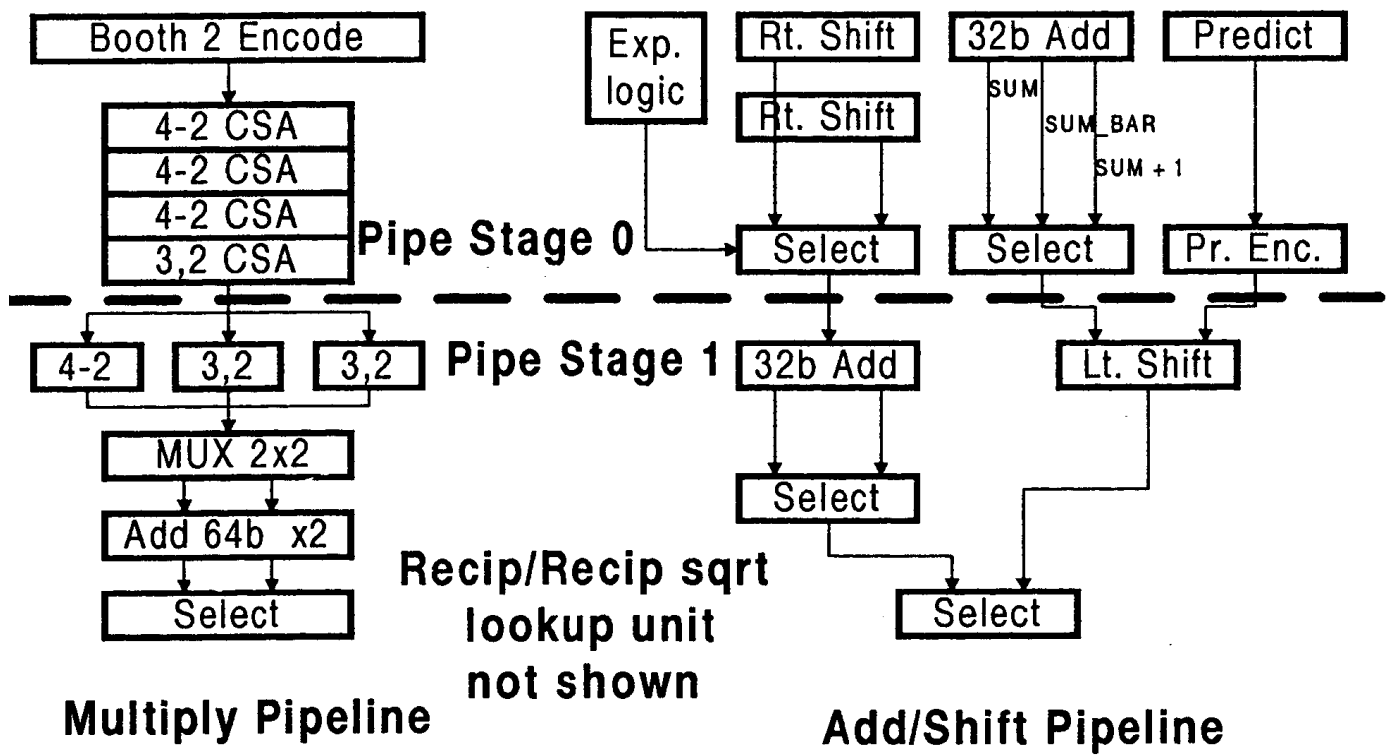
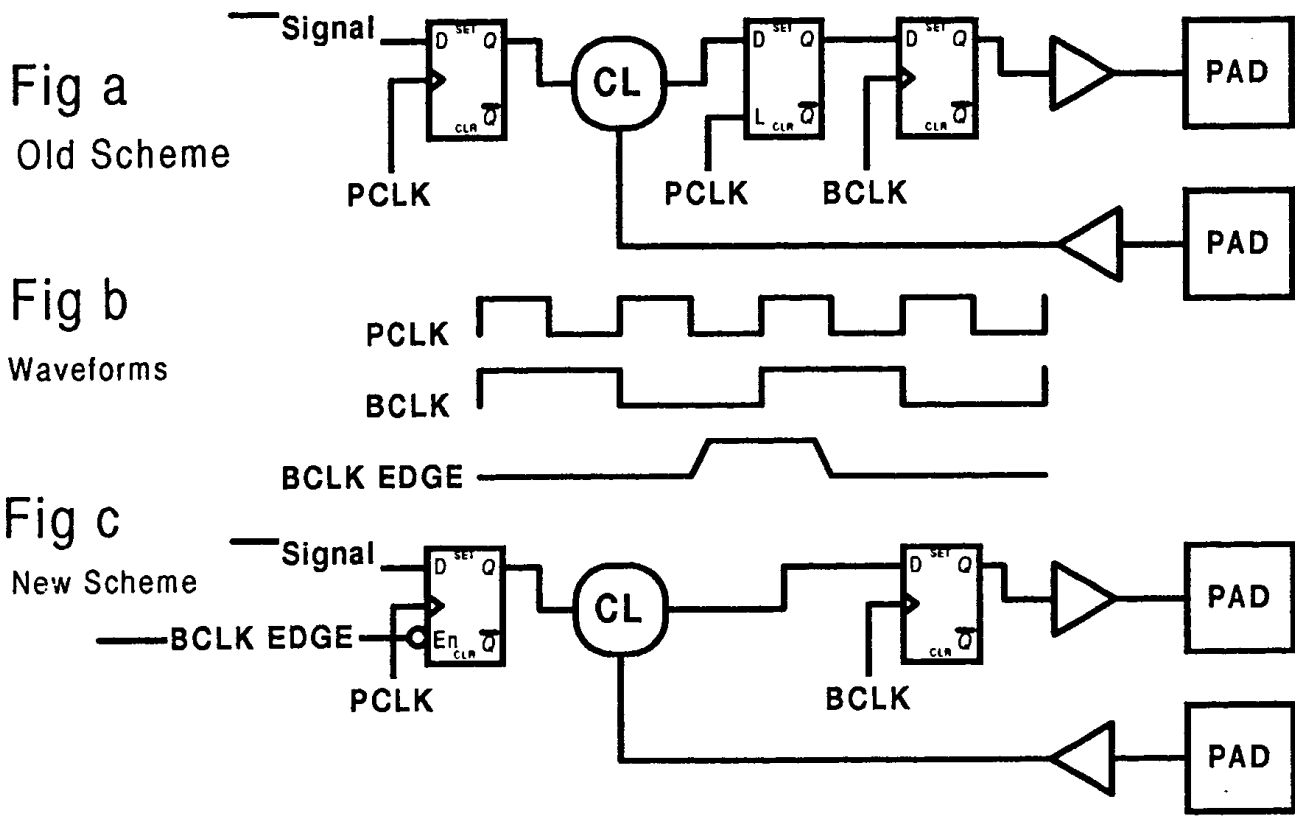


Figure 1: 3D Execution Units.



**Figure 2: Loop-back path logic.**

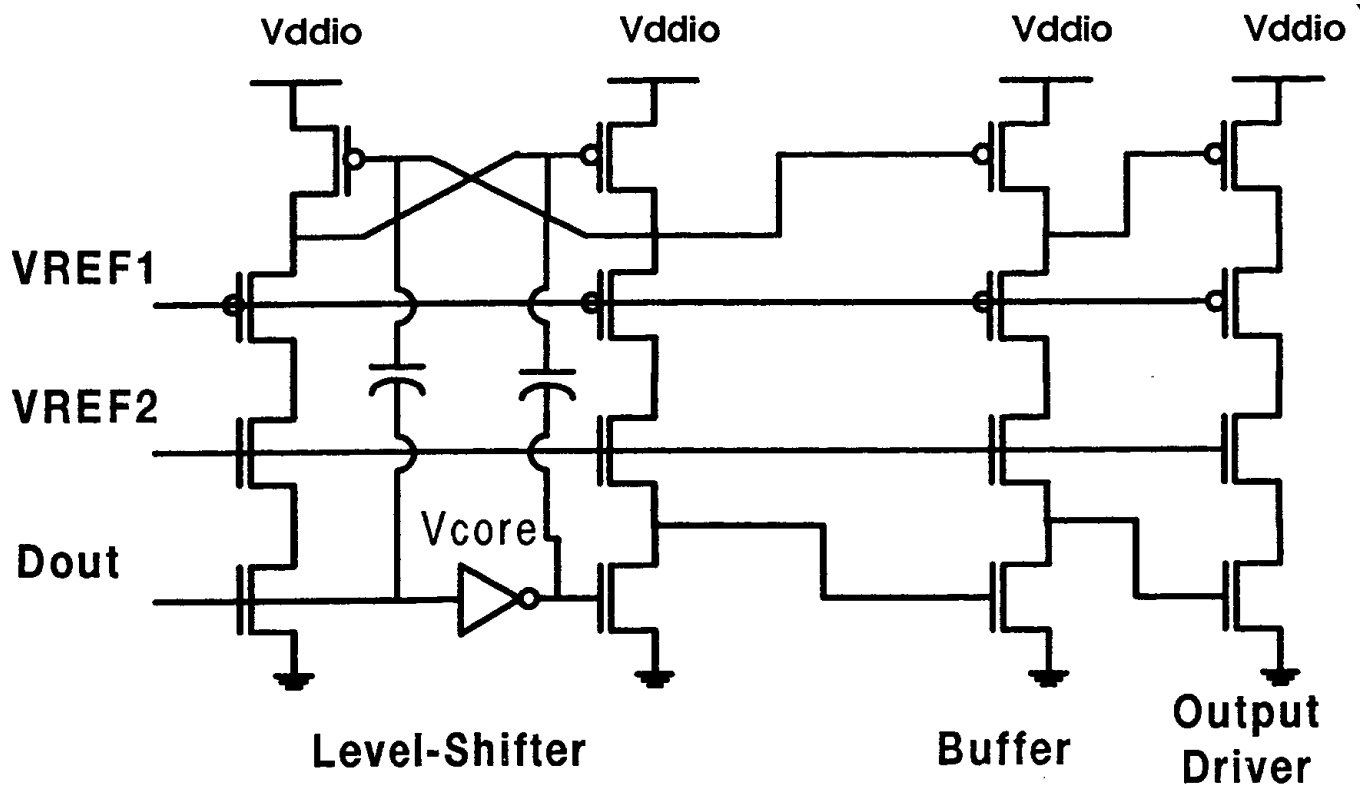


Figure 3: Protected output driver.

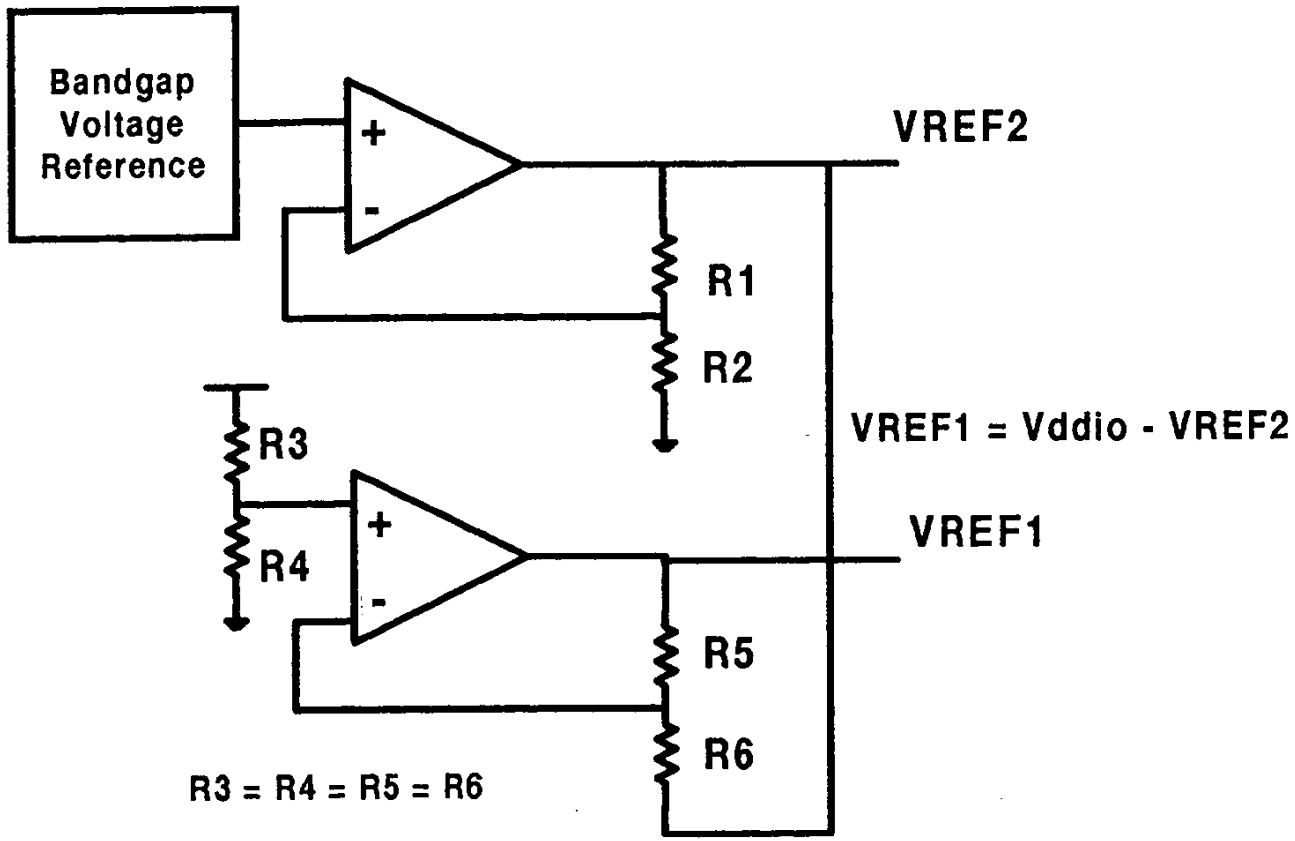
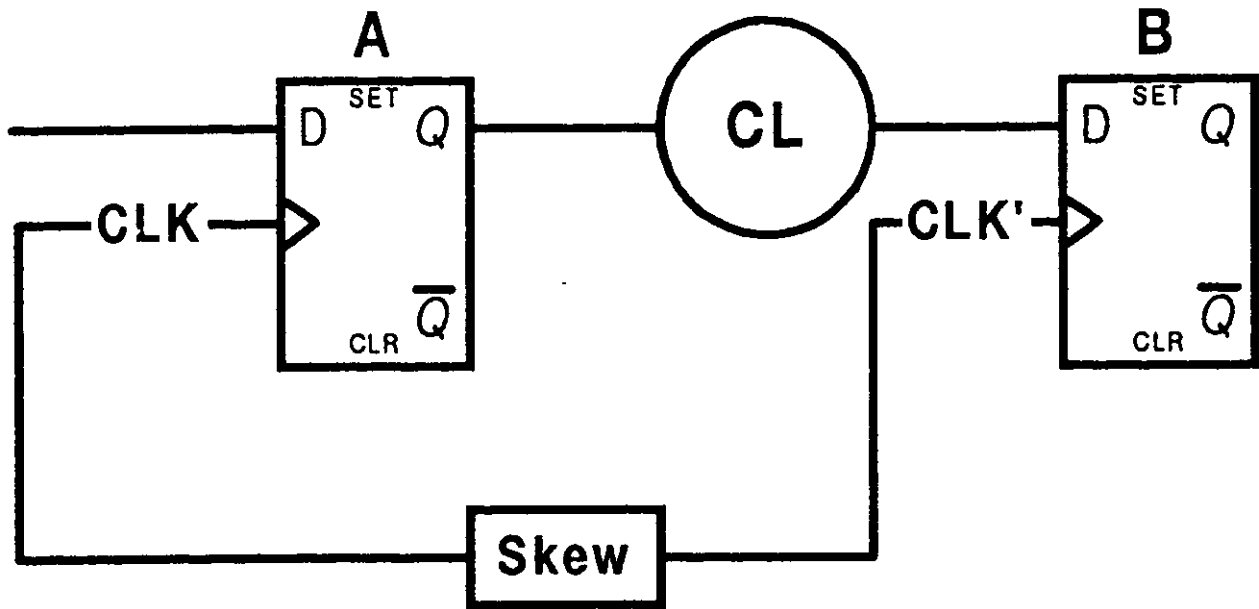


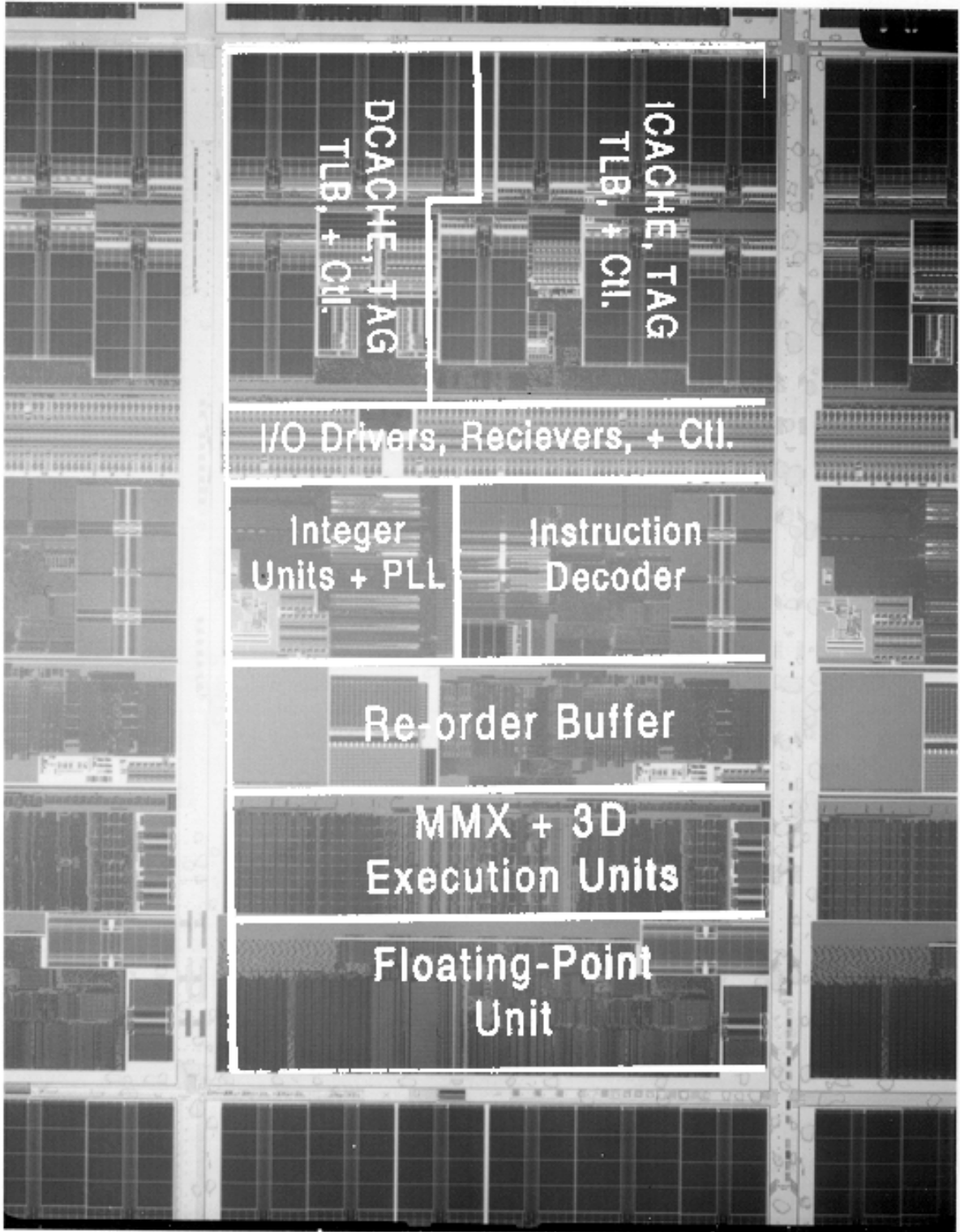
Figure 4: Reference voltage generation circuit.



$$T_{clk \rightarrow q, A} + T_{cl} > T_{skew} + T_{hold, B}$$

$$T_{skew} = T_{clk'} - T_{clk}$$

Figure 5: Clock vs. data race.



**Figure 6: Die micrograph.**



Gate oxide thickness	4.8nm
Polysilicon width/space	0.25/0.375nm
Local interconnect width/space	0.25/0.375nm
Metal 1 - 3 width/space	0.5/0.375nm
Metal 4 width/space	0.625/0.5nm
Metal 5 width/space	1.5/1.5nm

**Table 1: 0.25 $\mu$ m process dimensions.**